课程回顾

- 分治法求解：分解、解决、合并（举例：归并排序）
- 求解递归式（计算时间复杂度）：
  - 代入法：猜测解+数学归纳法证明（注意边界条件！）
    - 注意事项：做出好的猜测、细节修正、避免陷阱、变量代换

递归式求解——代入法 (续)

■注意事项3：避免陷阱

- ➢证明时渐近记号的使用易产生错误

例：$T(n) = 2T(\lfloor n/2 \rfloor) + n$

猜测$T(n)=O(n)$，此时“证明”$T(n)\leq cn$

$T(n) \leq 2(c\lfloor n/2 \rfloor) + n \leq cn + n = O(n)$ 错误！

必须证明$T(n)\leq cn$的精确形式！

递归式求解——代入法 (续)

■注意事项4：变量代换

- 有时进行变量代换能使未知递归式变为熟悉的形式

例：$T(n) = 2T(\lfloor\sqrt{n}\rfloor) + \lg n$

令m=lg$n$得：$T(2^m) = 2T(2^{m/2}) + m$ (考虑$\sqrt{n}$是整数的情形)

再令$S(m)$=$T(2^m)$得：$S(m) = 2S(m/2) + m$

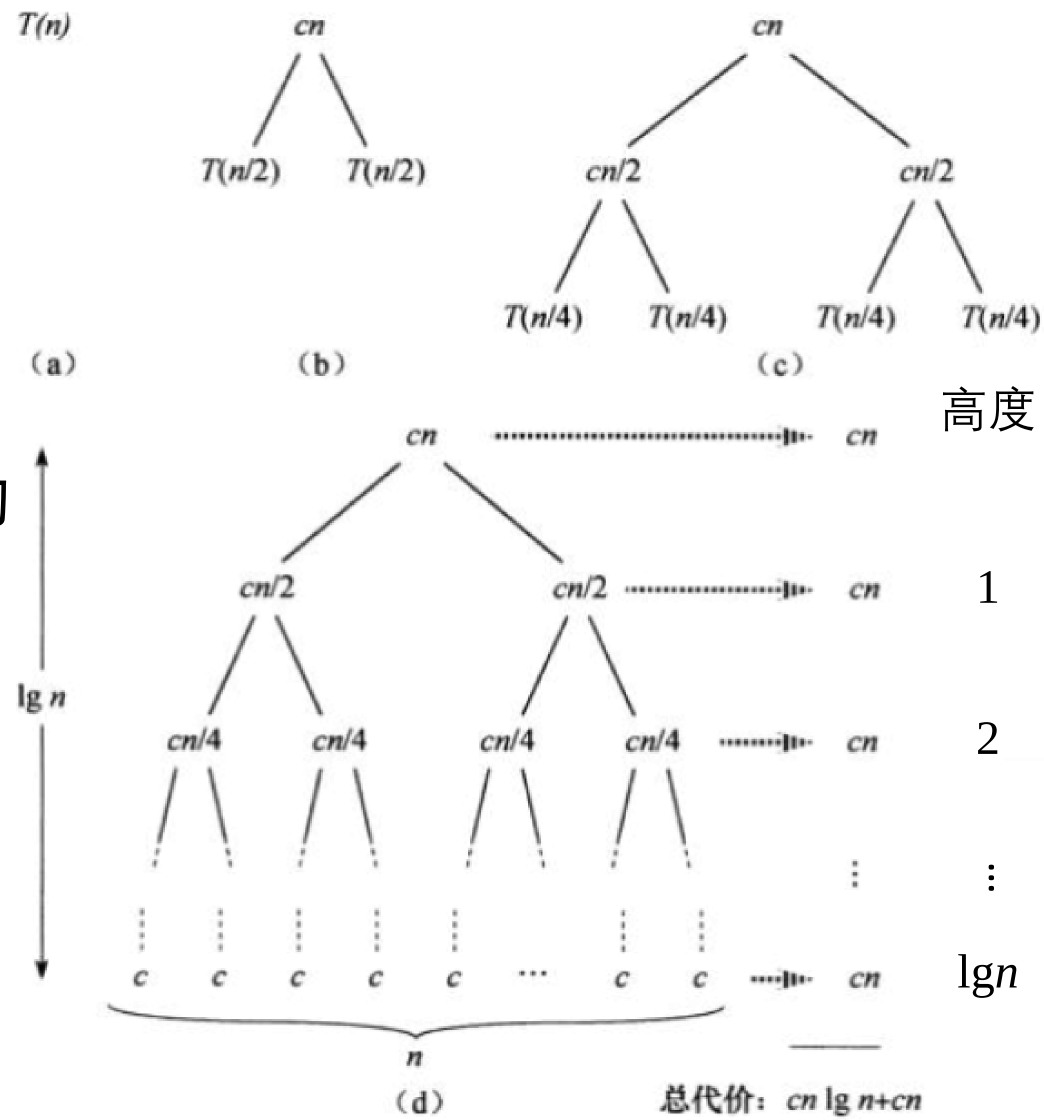
$S(m)=O(m\lg m) \Rightarrow T(n)=T(2^m)=S(m)=O(m\lg m)=O(\lg n\lg\lg n)$

递归式求解——递归树法

- 递归树中每个结点表示一个单一子问题的代价，子问题对应某次递归函数调用
- 树中每层的代价求和得到每层代价，将所有层的代价求和，得到所有层次递归调用总代价
- 递归树是展开过程的形象化，从$T(n)$逐步展开直到$T(1)$

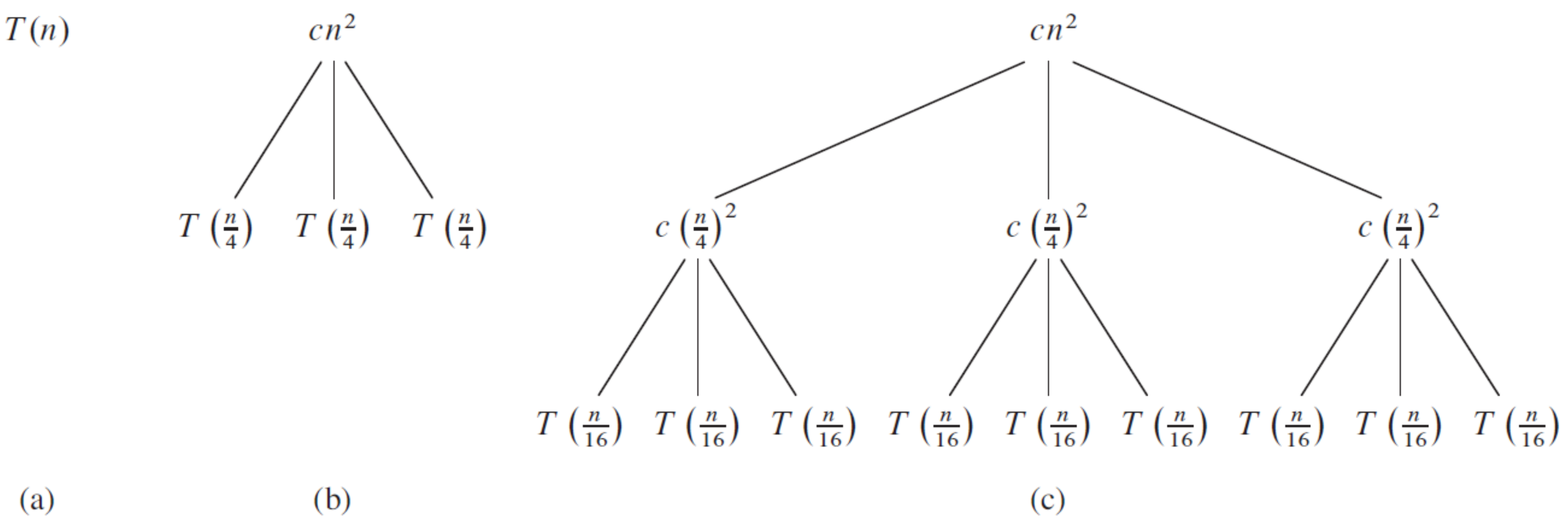
递归式求解——递归树法 (续)

- 例1：（教材p21）
  归并排序
  $T(n)=2T(n/2)+cn$
  （不妨设$n=2^k$）
  - 树高（从根到叶的最长简单路径长度）：$\lg n$
  - 总层数：$\lg n+1$
  - 每层代价：cn
  - 总代价：$cn\lg n+cn$



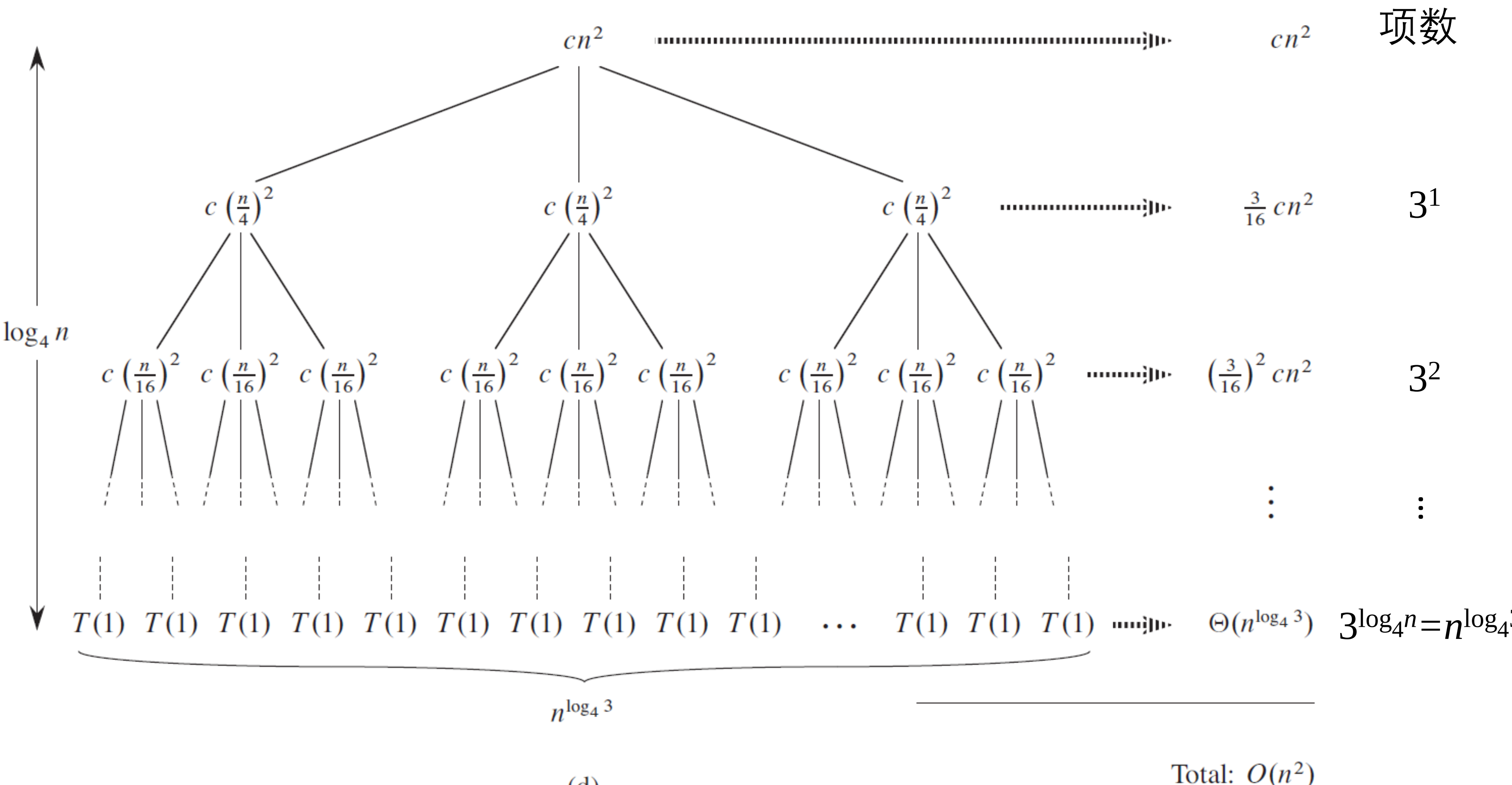
递归式求解——递归树法 (续)

■例2：（教材p51）$T(n) = 3T(\lfloor n/4 \rfloor) + cn^2$ （设n=4^k）



递归式求解——递归树法 (续)

■例2： （教材p51）$T(n) = 3T(\lfloor n/4 \rfloor) + cn^2$ （设$n=4^k$）



递归式求解——递归树法 (续)

■例2：（教材p51）$T(n) = 3T(\lfloor n/4 \rfloor) + cn^2$ （设n=4^k）

$$
\begin{aligned}
T(n) &= cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \ldots + \left(\frac{3}{16}\right)^{\log_4 n - 1} cn^2 + \Theta(n^{\log_4 3}) \\
&= \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
&= \frac{1 - (3/16)^{\log_4 n}}{13/16} cn^2 + \Theta(n^{\log_4 3}) \\
&< \frac{16}{13} cn^2 + \Theta(n^{\log_4 3}) \qquad // 0 < (3/16)^{\log_4 n} \le 1 \\
&= O(n^2)
\end{aligned}
$$

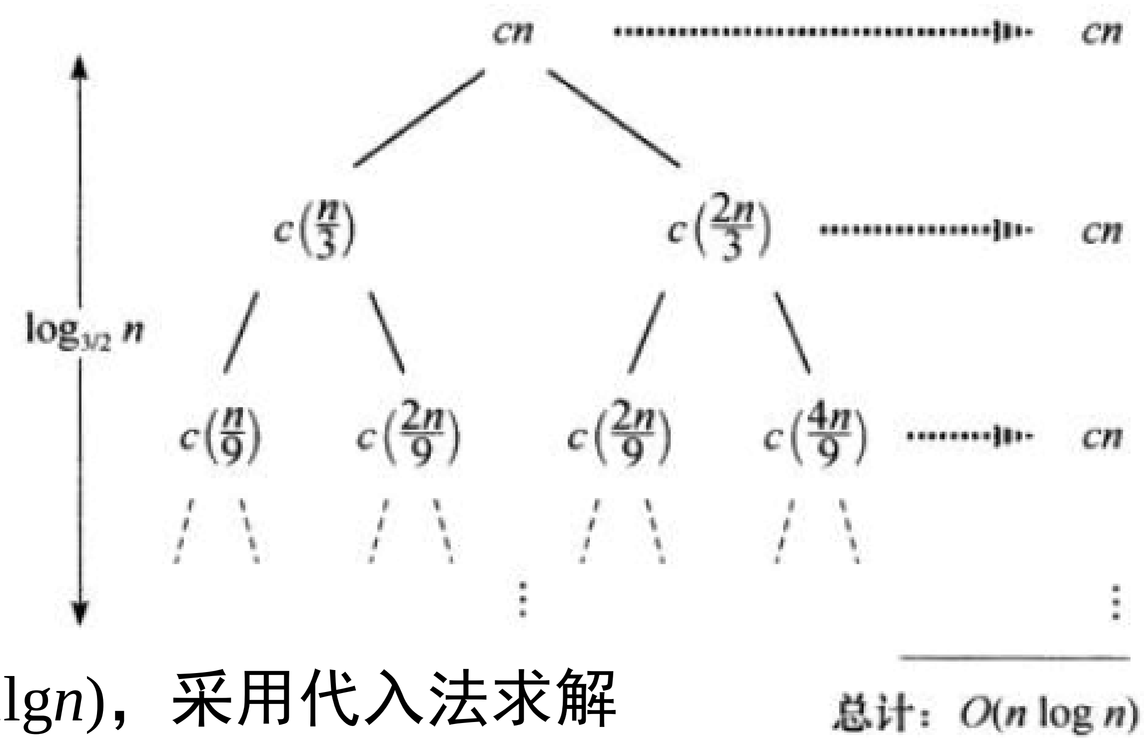
递归式求解——递归树法 (续)

■例3：（教材p52）更复杂的例子：树不一定是满二叉树，叶子深度不尽相同

$T(n)=T(n/3)+T(2n/3)+O(n)$

树高k满足：

$(2/3)^k \cdot n=1$



猜测$T(n)=O(n\lg n)$，采用代入法求解

递归式求解——主方法

- The master method，通用法，万能法
- 可迅速求解
  - $T(n)=aT(n/b)+f(n)$ //常数$a\geq 1$，$b>1$，$f(n)$为渐近正函数
  - 将规模为n的问题划分为a个子问题，每个子问题规模为n/b，每个子问题时间为$T(n/b)$，划分和合并的时间为$f(n)$
  - 注：n/b不一定为整数，实际中应当用$\lceil n/b \rceil$或$\lfloor n/b \rfloor$替换，但不影响渐近性质

递归式求解——主方法 (续)

■定理4.1（教材p53-54，主定理） 令a≥1和b>1是常数，$f(n)$是一个函数，$T(n)$是定义在非负整数上的递归式：

$$T(n) = aT(n/b) + f(n)$$

其中我们将n/b解释为$\lceil n/b \rceil$或$\lfloor n/b \rfloor$，有如下渐近界：

1. 若对某个常数ε>0有 $f(n) = O(n^{\log_b a-\varepsilon})$，则$T(n) = \Theta(n^{\log_b a})$
2. 若$f(n) = \Theta(n^{\log_b a})$，则$T(n) = \Theta(n^{\log_b a} \lg n)$
3. 若对某个常数ε>0有$f(n) = \Omega(n^{\log_b a+\varepsilon})$，且对某个常数$c$<1和所有足够大的$n$有$af(n/b) \leq cf(n)$，则$T(n) = \Theta(f(n))$

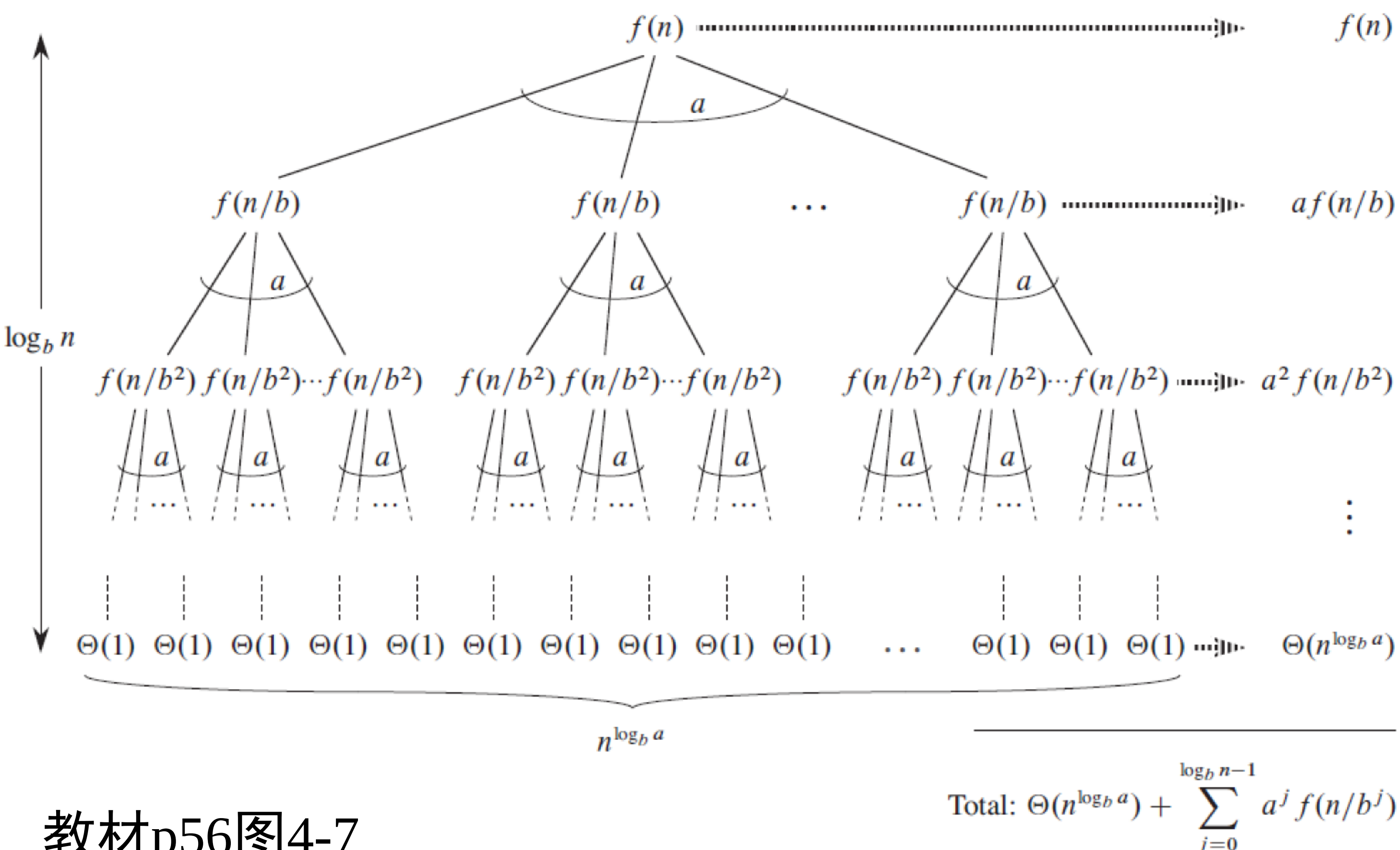
递归式求解——主方法 (续)

■定理意义：比较$f(n)$和$n^{\log_b a}$，直观上两函数较大者决定$T(n)$

1. $n^{\log_b a}$比$f(n)$大一个多项式因子n^{ε}：$T(n) = \Theta(n^{\log_b a})$
2. 两者相同，乘以对数因子lg$n$：
   $T(n) = \Theta(n^{\log_b a} \lg n) = \Theta(\lg n f(n))$
3. $f(n)$比$n^{\log_b a}$大一个多项式因子n^{ε}，以及满足“正则”条件：$T(n) = \Theta(f(n))$

■注：三种情况并未覆盖所有可能的$f(n)$，存在间隙

递归式求解——主方法 (续)



教材p56图4-7

递归式求解——主方法 (续)

$T(n) = aT(n/b) + f(n)$
子问题个数a，子问题规模n/b
求解与合并时间$f(n)$与$n^{\log_b a}$比较

■例1：

$$T(n) = 9T(n/3) + n$$

- a=9，b=3，$f(n)$=n，$n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$
- $f(n) = O(n^{\log_3 9-1}) = O(n)$, $\varepsilon = 1$
- $n^{\log_b a}$更大，主定理第1种情况：$T(n)=\Theta(n^2)$

递归式求解——主方法 (续)

$T(n) = aT(n/b) + f(n)$

子问题个数a，子问题规模n/b

求解与合并时间$f(n)$与$n^{\log_b a}$比较

■例2：

$$T(n) = T(2n/3) + 1$$

- a=1，b=3/2，$f(n)$=1，$n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = \Theta(1)$
- $f(n) = 1 = \Theta(1)$
- $n^{\log_b a}$与$f(n)$同级别，主定理第2种情况：$T(n)=\Theta(\lg n)$

递归式求解——主方法 (续)

$T(n) = aT(n/b) + f(n)$

子问题个数*a*，子问题规模*n*/*b*

求解与合并时间*f*(*n*)与$n^{\log_b a}$比较

■例3：

$$T(n) = 3T(n/4) + n\lg n$$

- ➢*a*=3，*b*=4，*f*(*n*)=nlg*n*，$n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$
- ➢$f(n) = n\lg n = \Omega(n) = \Omega(n^{\log_4 3+\epsilon})$，$\varepsilon \approx 0.2$
- ➢*c*=3/4且*n*足够大时，*af*(*n*/*b*)=3(*n*/4)lg(*n*/4)≤(3/4)nlg*n*=*cf*(*n*)
- ➢*f*(*n*)更大，主定理第3种情况：*T*(*n*)=Θ(*n*lg*n*)

递归式求解——主方法 (续)

$T(n) = aT(n/b) + f(n)$
子问题个数a，子问题规模n/b
求解与合并时间$f(n)$与$n^{\log_b a}$比较

■例4：

$$T(n) = 2T(n/2) + n\lg n$$

- a=2，b=2，$f(n)$=nlgn，$n^{\log_b a} = n^{\log_2 2} = \Theta(n)$
- $f(n) = n\lg n = \Omega(n)$，但找不到ε使得$n\lg n = \Omega(n^{1+\varepsilon})$

$$\lim_{n\to\infty} \frac{n^{\varepsilon}}{\lg n} = \infty, \quad \varepsilon > 0$$

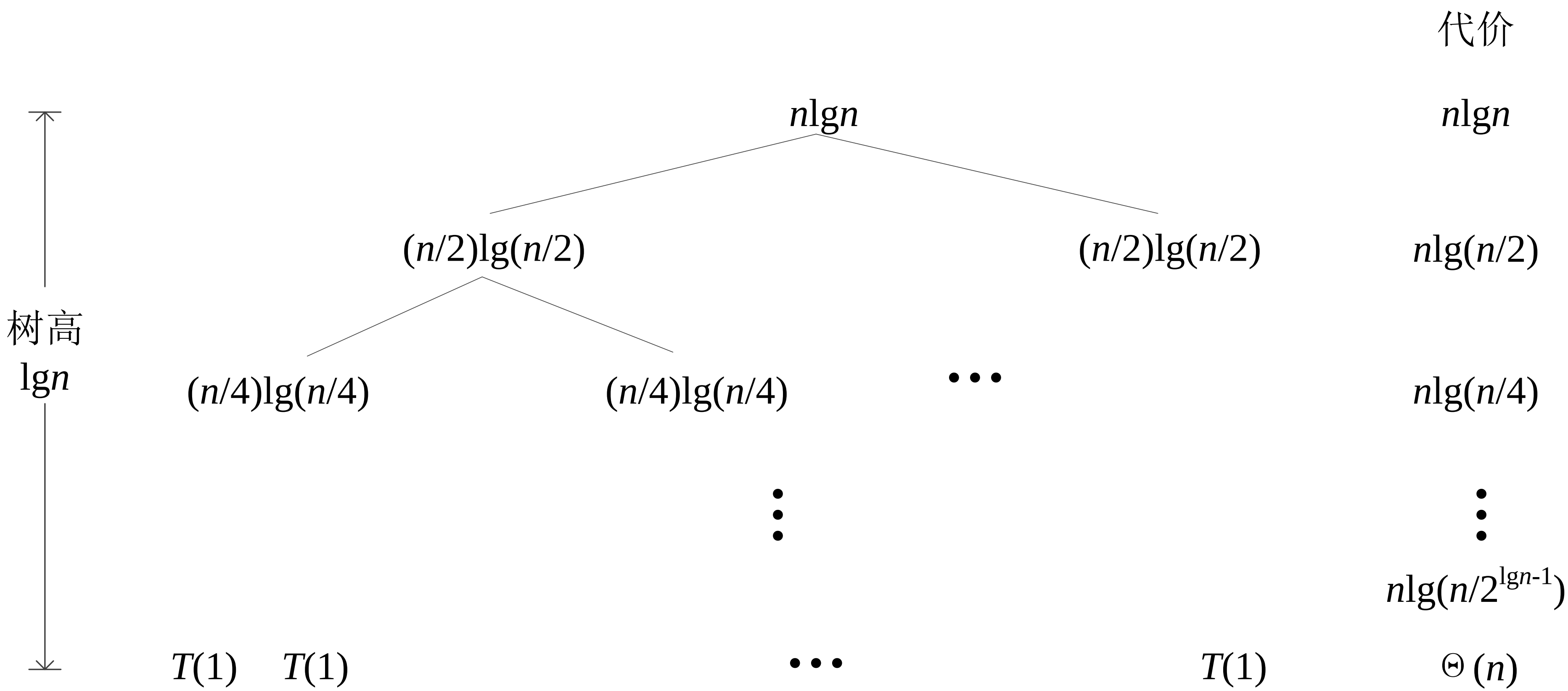
- 该递归式落入了情况2和3的间隙，无法使用主方法

如何求解？

递归式求解——主方法 (续)

■例4：（递归树法）

$$T(n) = 2T(n/2) + n\lg n$$



递归式求解——主方法 (续)

■例4：（递归树法）

$$T(n) = 2T(n/2) + n \lg n$$

$$\begin{aligned} T(n) &= \sum_{i=0}^{\lg n - 1} n \lg \frac{n}{2^i} + \Theta(n) \\ &= n \sum_{i=0}^{\lg n - 1} (\lg n - i) + \Theta(n) \\ &= n \lg^2 n - \frac{n(\lg^2 n - \lg n)}{2} + \Theta(n) \\ &= \frac{1}{2} n \lg^2 n + \frac{1}{2} n \lg n + \Theta(n) \\ &= \Theta(n \lg^2 n) \end{aligned}$$

递归式求解例

4-3 （更多的递归式例子） 对下列每个递归式，给出 $T(n)$的渐近上界和下界。假定对足够小的 n，$T(n)$是常数。给出尽量紧确的界，并验证其正确性。

a. $T(n)=4T(n/3)+n\lg n$

b. $T(n)=3T(n/3)+n/\lg n$

c. $T(n)=4T(n/2)+n^2\sqrt{n}$

d. $T(n)=3T(n/3-2)+n/2$

e. $T(n)=2T(n/2)+n/\lg n$

f. $T(n)=T(n/2)+T(n/4)+T(n/8)+n$

g. $T(n)=T(n-1)+1/n$

h. $T(n)=T(n-1)+\lg n$

j. $T(n)=\sqrt{n}T(\sqrt{n})+n$

递归式求解例 (续)

a. $T(n) = 4T(n/3) + n\lg n$

解：a=4, b=3, $n^{\log_b a} = n^{\log_3 4}$, $f(n)$=$n\lg n$

$f(n) = O(n^{\log_3 4-\varepsilon})$, 其中$0 < \varepsilon < \log_3 \frac{4}{3}$即可

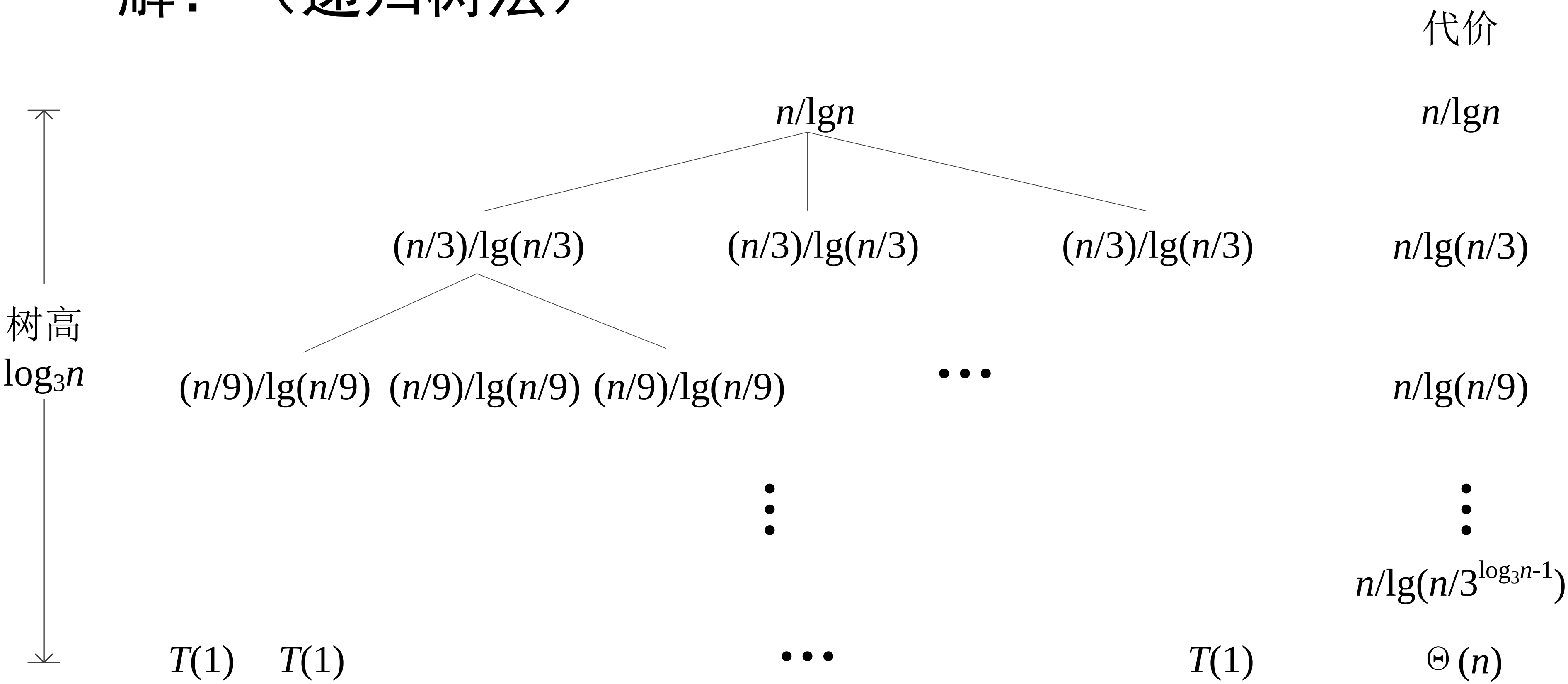
主定理第1种情况，$T(n) = \Theta(n^{\log_3 4})$

递归式求解例 (续)

b. $T(n) = 3T(n/3) + n/\lg n$

主定理无法求解

解：（递归树法）



递归式求解例 (续)

b. $T(n) = 3T(n/3) + n/\lg n$

解：
$$T(n) = \frac{n}{\lg n} + \frac{n}{\lg(n/3)} + \ldots + \frac{n}{\lg(n/3^{\log_3 n - 1})} + \Theta(n)$$
$$= \sum_{i=0}^{\log_3 n - 1} \frac{n}{\lg(n/3^i)} + \Theta(n)$$
$$= n \sum_{i=0}^{\log_3 n - 1} \frac{1}{\lg n - i \lg 3} + \Theta(n)$$
$$= \frac{n}{\lg 3} \sum_{i=0}^{\log_3 n - 1} \frac{1}{\log_3 n - i} + \Theta(n)$$
$$= \frac{n}{\lg 3} \left(1 + \frac{1}{2} + \ldots + \frac{1}{\log_3 n}\right) + \Theta(n) = O(n \lg \lg n)$$

$n \longrightarrow \infty$时为调和级数，约为$\ln\log_3 n$

递归式求解例 (续)

c. $T(n) = 4T(n/2) + n^2\sqrt{n}$

解： a=4, b=2, $n^{\log_b a} = n^{\log_2 4} = n^2$

$f(n) = n^{5/2} = \Omega(n^{2+1/2}), \varepsilon = 1/2$

$c = \sqrt{2}/2$ 且n足够大时,

$af(n/b) = 4(n/2)^2\sqrt{n/2} = \frac{\sqrt{2}}{2}n^2\sqrt{n} \le cf(n)$

主定理第3种情况, $T(n) = \Theta(n^{5/2})$

递归式求解例 (续)

d. $T(n) = 3T(n/3-2) + n/2$

当$n \to \infty$时$n/3-2 \approx n/3$，即相当于求解

$T(n) = 3T(n/3) + n/2$

满足主定理第2种情况，$T(n) = \Theta(n\lg n)$

严格证明需进一步通过代入法求解

以下使用代入法证明$T(n)=O(n\lg n)$

递归式求解例 (续)

d. $T(n) = \begin{cases} 1, & n = 1..6, \\ 3T(\lceil n/3 - 2 \rceil) + n/2, & n \geq 7. \end{cases}$

解：猜想$T(n) = O(n\lg n)$，需证明恰当选择常数c>0，有$T(n) \leq cn\lg n$

$T(n) \leq 3c(\boxed{n/3 - 2 + 1})\lg(\boxed{n/3 - 2 + 1}) + n/2 \leq cn\lg(n/3) + n/2$
$= cn\lg n - cn\lg 3 + n/2 \leq cn\lg n$

$$x - 1 < \lfloor x \rfloor \leq x \leq \boxed{\lceil x \rceil < x + 1}, \quad \forall x \in \mathbb{R}$$

$c \geq 1/(2\lg 3)$即可，可取c=1/2。当n=1时，$T(1)>cn\lg n$=0不满足条件，此时选择n_0=2，即上述不等式成立条件为$\lceil n/3 - 2 \rceil \geq 2$，即$n$≥10，因此$n$=2..9为归纳边界条件。显然$n$=2..6时，$T(n)$=1≤$cn\lg n$，
$T(7)=3T(1)+7/2=13/2\leq(7\lg 7)/2$，$T(8)=3T(1)+4=7\leq 12$，
$T(9)=3T(1)+9/2=15/2\leq(9\lg 9)/2$，均成立。

因此，取c=1/2, n_0=2，当$n \geq n_0$时，有$0 \leq T(n) \leq cn\lg n$，即$T(n) = O(n\lg n)$

递归式求解例 (续)

e. $T(n) = 2T(n/2) + n/\lg n$ （同b $T(n) = O(n\lg\lg n)$）

f. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

思考：$n/2 + n/4 + n/8 = 7n/8 < n$，猜想$T(n) = O(n)$

解1：（代入法）猜测解$T(n)=O(n)$，需要证明恰当选择常数$c>0$，有$T(n)\leq cn$

$T(n) \leq cn/2 + cn/4 + cn/8 + n = (7c/8 + 1)n \leq cn$

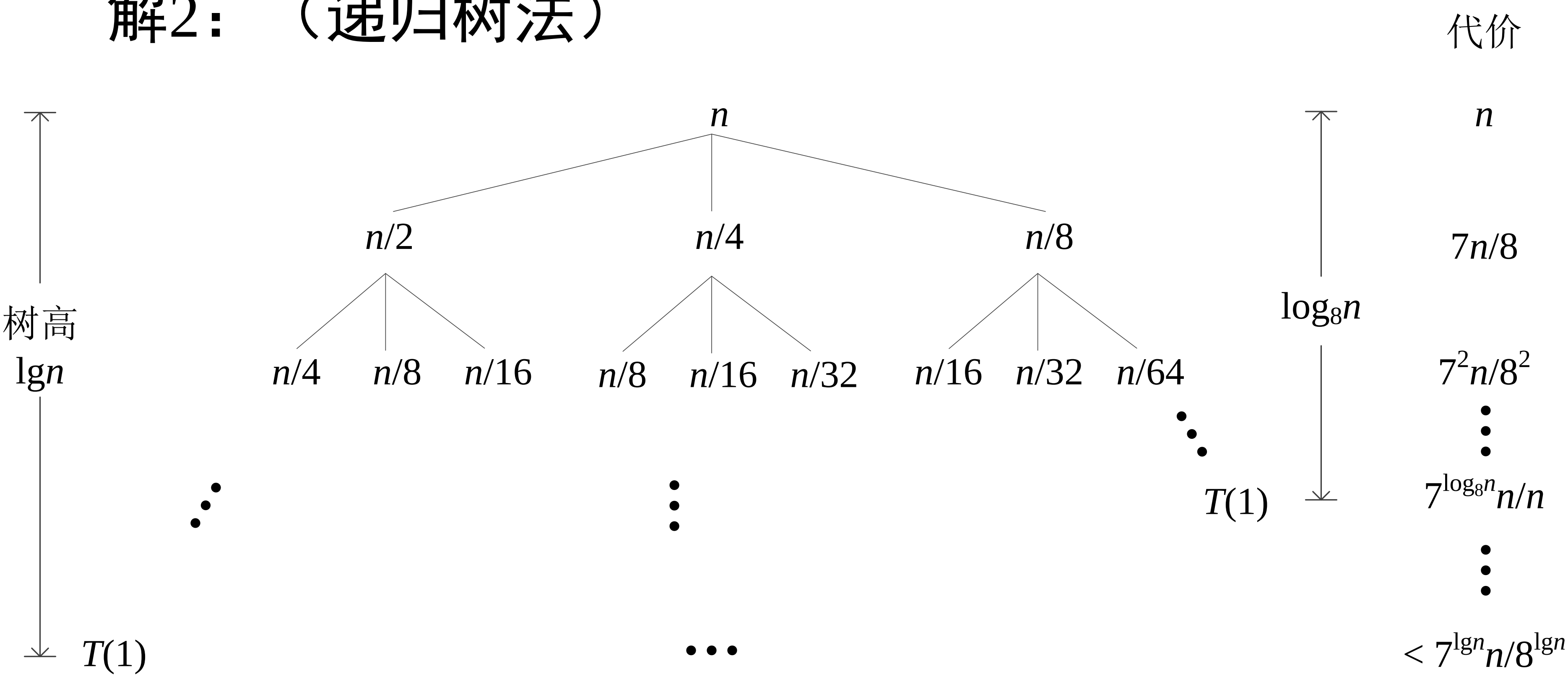
$c\geq 8$即可

若题目给出递归边界条件，则还需证明边界情况

递归式求解例 (续)

f. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

解2：（递归树法）



递归式求解例 (续)

f. $T(n) = T(n/2) + T(n/4) + T(n/8) + n$

解2：（递归树法）

$$
\begin{aligned}
T(n) &< n + \frac{7n}{8} + \frac{7^2 n}{8^2} + \ldots + \frac{7^{\lg n} n}{8^{\lg n}} \\
&= n \sum_{i=0}^{\lg n} (\frac{7}{8})^i \\
&= \frac{1-(7/8)^{\lg n+1}}{1-7/8} n \\
&< 8n \\
&= O(n)
\end{aligned}
$$

递归式求解例 (续)

g. $T(n) = T(n-1) + 1/n$

（递归树法，调和级数，$T(n) = O(\lg n)$）

h. $T(n) = T(n-1) + \lg n$

（递归树法，$T(n) = O(\lg(n!)) = O(n\lg n)$）